EXHIBIT C

ANNALS OF ARTIFICIAL INTELLIGENCE

CHATGPT IS A BLURRY JPEG OF THE WEB

OpenAI's chatbot offers paraphrases, whereas Google offers quotes. Which do we prefer?

By **Ted Chiang**
February 9, 2023

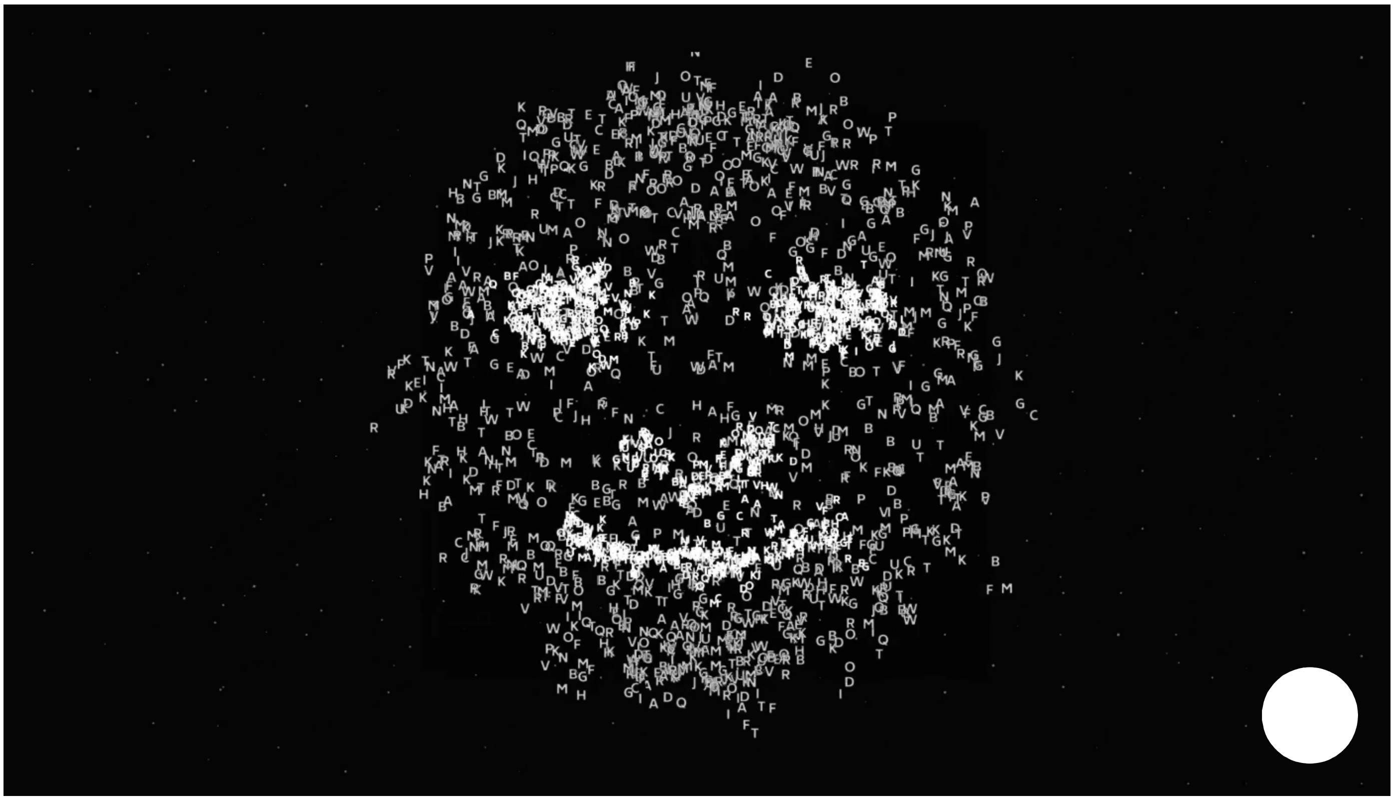

Illustration by Vivek Thakker

Save this story

In 2013, workers at a German construction company noticed something odd about their Xerox photocopier: when they made a copy of the floor plan of a house, the copy differed from the original in a subtle but significant way. In the original floor plan, each of the house's three rooms was accompanied by a

rectangle specifying its area: the rooms were 14.13, 21.11, and 17.42 square metres, respectively. However, in the photocopy, all three rooms were labelled as being 14.13 square metres in size. The company contacted the computer scientist David Kriesel to investigate this seemingly inconceivable result. They needed a computer scientist because a modern Xerox photocopier doesn't use the physical xerographic process popularized in the nineteen-sixties. Instead, it scans the document digitally, and then prints the resulting image file. Combine that with the fact that virtually every digital image file is compressed to save space, and a solution to the mystery begins to suggest itself.

Compressing a file requires two steps: first, the encoding, during which the file is converted into a more compact format, and then the decoding, whereby the process is reversed. If the restored file is identical to the original, then the compression process is described as lossless: no information has been discarded. By contrast, if the restored file is only an approximation of the original, the compression is described as lossy: some information has been discarded and is now unrecoverable. Lossless compression is what's typically used for text files and computer programs, because those are domains in which even a single incorrect character has the potential to be disastrous. Lossy compression is often used for photos, audio, and video in situations in which absolute accuracy isn't essential. Most of the time, we don't notice if a picture, song, or movie isn't perfectly reproduced. The loss in fidelity becomes more perceptible only as files are squeezed very tightly. In those cases, we notice what are known as compression artifacts: the fuzziness of the smallest JPEG and MPEG images, or the tinny sound of low-bit-rate MP3s.

Xerox photocopiers use a lossy compression format known as JBIG2, designed for use with black-and-white images. To save space, the copier identifies similar-looking regions in the image and stores a single copy for all of them; when the file is decompressed, it uses that copy repeatedly to reconstruct the image. It turned out that the photocopier had judged the labels specifying the area of the rooms to be similar enough that it needed to store only one of them—14.13—and it reused that one for all three rooms when printing the floor plan.

The fact that Xerox photocopiers use a lossy compression format instead of a lossless one isn't, in itself, a problem. The problem is that the photocopiers were degrading the image in a subtle way, in which the compression artifacts weren't immediately recognizable. If the photocopier simply produced blurry printouts, everyone would know that they weren't accurate reproductions of the originals. What led to problems was the fact that the photocopier was producing numbers that were readable but incorrect; it made the copies seem accurate when they weren't. (In 2014, Xerox released a patch to correct this issue.)

I think that this incident with the Xerox photocopier is worth bearing in mind today, as we consider OpenAI's ChatGPT and other similar programs, which A.I. researchers call large language models. The resemblance between a photocopier and a large language model might not be immediately apparent—but consider the

following scenario. Imagine that you're about to lose your access to the Internet forever. In preparation, you plan to create a compressed copy of all the text on the Web, so that you can store it on a private server. Unfortunately, your private server has only one per cent of the space needed; you can't use a lossless compression algorithm if you want everything to fit. Instead, you write a lossy algorithm that identifies statistical regularities in the text and stores them in a specialized file format. Because you have virtually unlimited computational power to throw at this task, your algorithm can identify extraordinarily nuanced statistical regularities, and this allows you to achieve the desired compression ratio of a hundred to one.

Now, losing your Internet access isn't quite so terrible; you've got all the information on the Web stored on your server. The only catch is that, because the text has been so highly compressed, you can't look for information by searching for an exact quote; you'll never get an exact match, because the words aren't what's being stored. To solve this problem, you create an interface that accepts queries in the form of questions and responds with answers that convey the gist of what you have on your server.

What I've described sounds a lot like ChatGPT, or most any other large language model. Think of ChatGPT as a blurry JPEG of all the text on the Web. It retains much of the information on the Web, in the same way that a JPEG retains much of the information of a higher-resolution image, but, if you're looking for an exact sequence of bits, you won't find it; all you will ever get is an approximation. But, because the approximation is presented in the form of grammatical text, which ChatGPT excels at creating, it's usually acceptable. You're still looking at a blurry JPEG, but the blurriness occurs in a way that doesn't make the picture as a whole look less sharp.

This analogy to lossy compression is not just a way to understand ChatGPT's facility at repackaging information found on the Web by using different words. It's also a way to understand the "hallucinations," or nonsensical answers to factual

questions, to which large language models such as ChatGPT are all too prone. These hallucinations are compression artifacts, but—like the incorrect labels generated by the Xerox photocopier—they are plausible enough that identifying them requires comparing them against the originals, which in this case means either the Web or our own knowledge of the world. When we think about them this way, such hallucinations are anything but surprising; if a compression algorithm is designed to reconstruct text after ninety-nine per cent of the original has been discarded, we should expect that significant portions of what it generates will be entirely fabricated.

This analogy makes even more sense when we remember that a common technique used by lossy compression algorithms is interpolation—that is, estimating what's missing by looking at what's on either side of the gap. When an image program is displaying a photo and has to reconstruct a pixel that was lost during the compression process, it looks at the nearby pixels and calculates the average. This is what ChatGPT does when it's prompted to describe, say, losing a sock in the dryer using the style of the Declaration of Independence: it is taking two points in "lexical space" and generating the text that would occupy the location between them. ("When in the Course of human events, it becomes necessary for one to separate his garments from their mates, in order to maintain the cleanliness and order thereof. . . .") ChatGPT is so good at this form of interpolation that people find it entertaining: they've discovered a "blur" tool for paragraphs instead of photos, and are having a blast playing with it.

Given that large language models like ChatGPT are often extolled as the cutting edge of artificial intelligence, it may sound dismissive—or at least deflating—to describe them as lossy text-compression algorithms. I do think that this perspective offers a useful corrective to the tendency to anthropomorphize large language models, but there is another aspect to the compression analogy that is worth considering. Since 2006, an A.I. researcher named Marcus Hutter has offered a cash reward—known as the Prize for Compressing Human Knowledge,

or the Hutter Prize—to anyone who can losslessly compress a specific one-gigabyte snapshot of Wikipedia smaller than the previous prize-winner did. You have probably encountered files compressed using the zip file format. The zip format reduces Hutter's one-gigabyte file to about three hundred megabytes; the most recent prize-winner has managed to reduce it to a hundred and fifteen megabytes. This isn't just an exercise in smooshing. Hutter believes that better text compression will be instrumental in the creation of human-level artificial intelligence, in part because the greatest degree of compression can be achieved by understanding the text.

To grasp the proposed relationship between compression and understanding, imagine that you have a text file containing a million examples of addition, subtraction, multiplication, and division. Although any compression algorithm could reduce the size of this file, the way to achieve the greatest compression ratio would probably be to derive the principles of arithmetic and then write the code for a calculator program. Using a calculator, you could perfectly reconstruct not just the million examples in the file but any other example of arithmetic that you might encounter in the future. The same logic applies to the problem of compressing a slice of Wikipedia. If a compression program knows that force equals mass times acceleration, it can discard a lot of words when compressing the pages about physics because it will be able to reconstruct them. Likewise, the more the program knows about supply and demand, the more words it can discard when compressing the pages about economics, and so forth.

Large language models identify statistical regularities in text. Any analysis of the text of the Web will reveal that phrases like "supply is low" often appear in close proximity to phrases like "prices rise." A chatbot that incorporates this correlation might, when asked a question about the effect of supply shortages, respond with an answer about prices increasing. If a large language model has compiled a vast number of correlations between economic terms—so many that it can offer plausible responses to a wide variety of questions—should we say that it actually

understands economic theory? Models like ChatGPT aren't eligible for the Hutter Prize for a variety of reasons, one of which is that they don't reconstruct the original text precisely—i.e., they don't perform lossless compression. But is it possible that their lossy compression nonetheless indicates real understanding of the sort that A.I. researchers are interested in?

Let's go back to the example of arithmetic. If you ask GPT-3 (the large-language model that ChatGPT was built from) to add or subtract a pair of numbers, it almost always responds with the correct answer when the numbers have only two digits. But its accuracy worsens significantly with larger numbers, falling to ten per cent when the numbers have five digits. Most of the correct answers that GPT-3 gives are not found on the Web—there aren't many Web pages that contain the text "245 + 821," for example—so it's not engaged in simple memorization. But, despite ingesting a vast amount of information, it hasn't been able to derive the principles of arithmetic, either. A close examination of GPT-3's incorrect answers suggests that it doesn't carry the "1" when performing arithmetic. The Web certainly contains explanations of carrying the "1," but GPT-3 isn't able to incorporate those explanations. GPT-3's statistical analysis of examples of arithmetic enables it to produce a superficial approximation of the real thing, but no more than that.

Given GPT-3's failure at a subject taught in elementary school, how can we explain the fact that it sometimes appears to perform well at writing college-level essays? Even though large language models often hallucinate, when they're lucid they sound like they actually understand subjects like economic theory. Perhaps arithmetic is a special case, one for which large language models are poorly suited. Is it possible that, in areas outside addition and subtraction, statistical regularities in text actually *do* correspond to genuine knowledge of the real world?
I think there's a simpler explanation. Imagine what it would look like if ChatGPT were a lossless algorithm. If that were the case, it would always answer questions by providing a verbatim quote from a relevant Web page. We would probably

regard the software as only a slight improvement over a conventional search engine, and be less impressed by it. The fact that ChatGPT rephrases material from the Web instead of quoting it word for word makes it seem like a student expressing ideas in her own words, rather than simply regurgitating what she's read; it creates the illusion that ChatGPT understands the material. In human students, rote memorization isn't an indicator of genuine learning, so ChatGPT's inability to produce exact quotes from Web pages is precisely what makes us think that it has learned something. When we're dealing with sequences of words, lossy compression looks smarter than lossless compression.

A lot of uses have been proposed for large language models. Thinking about them as blurry JPEGs offers a way to evaluate what they might or might not be well suited for. Let's consider a few scenarios.

Can large language models take the place of traditional search engines? For us to have confidence in them, we would need to know that they haven't been fed propaganda and conspiracy theories—we'd need to know that the JPEG is capturing the right sections of the Web. But, even if a large language model includes only the information we want, there's still the matter of blurriness. There's a type of blurriness that is acceptable, which is the re-stating of information in different words. Then there's the blurriness of outright fabrication, which we consider unacceptable when we're looking for facts. It's not clear that it's technically possible to retain the acceptable kind of blurriness while eliminating the unacceptable kind, but I expect that we'll find out in the near future.

Even if it is possible to restrict large language models from engaging in fabrication, should we use them to generate Web content? This would make sense only if our goal is to repackage information that's already available on the Web. Some companies exist to do just that—we usually call them content mills. Perhaps the blurriness of large language models will be useful to them, as a way of avoiding copyright infringement. Generally speaking, though, I'd say that

anything that's good for content mills is not good for people searching for information. The rise of this type of repackaging is what makes it harder for us to find what we're looking for online right now; the more that text generated by large language models gets published on the Web, the more the Web becomes a blurrier version of itself.

There is very little information available about OpenAI's forthcoming successor to ChatGPT, GPT-4. But I'm going to make a prediction: when assembling the vast amount of text used to train GPT-4, the people at OpenAI will have made every effort to exclude material generated by ChatGPT or any other large language model. If this turns out to be the case, it will serve as unintentional confirmation that the analogy between large language models and lossy compression is useful. Repeatedly resaving a JPEG creates more compression artifacts, because more information is lost every time. It's the digital equivalent of repeatedly making photocopies of photocopies in the old days. The image quality only gets worse.

Indeed, a useful criterion for gauging a large language model's quality might be the willingness of a company to use the text that it generates as training material for a new model. If the output of ChatGPT isn't good enough for GPT-4, we might take that as an indicator that it's not good enough for us, either. Conversely, if a model starts generating text so good that it can be used to train new models, then that should give us confidence in the quality of that text. (I suspect that such an outcome would require a major breakthrough in the techniques used to build these models.) If and when we start seeing models producing output that's as good as their input, then the analogy of lossy compression will no longer be applicable.

Can large language models help humans with the creation of original writing? To answer that, we need to be specific about what we mean by that question. There is a genre of art known as Xerox art, or photocopy art, in which artists use the distinctive properties of photocopiers as creative tools. Something along those